Amendments to the Specification

i. Please add the BRIEF DESCRIPTION OF DRAWINGS section into the originally filed Specification as follows to reflect addition of the figures:

BRIEF DESCRIPTION OF DRAWINGS

FIG. 1A is a flowchart of an example embodiment of components of an information transformation software engine.

FIG. 1B is a mixed data and logic flowchart of an alternative view of components of an information transformation software engine.

FIG. 2 is a flowchart of another example embodiment of components of an information transformation software engine.

ii. Please amend the originally filed Specification as follows to reflect addition of the figures:

Amend the paragraph in the Specification starting from page 2, line 17 of the application to page 3, line 17 as follows:

As shown in the in FIG. 1A, FIG. 1B and FIG.2, example embodiments of components of an information transformation software engine are as follows: ~~The basic goals of the nSource system are as follows:~~

- Provide a software system that maximizes the (re)utility of software components that have general application. *Corollaries*:

  1. Divide (102) as many existing software processes (101) found in familiar functional domains into a normalized disassociated group of algorithms

(103)

2. Encapsulate (104) these with a unified configuration mechanism (130) that incorporates an easy-to-understand metaphor that is multi-modal (ie, easy to ingest from and dump to different formats/mediums).
3. Construct (106) a unified communications mechanism for these components (120) to employ within their own domain that is also multi-modal (ie, easy to mate with event-driven information from other applicationslhosts).

- Give both the software engineer and end user the tools to efficiently (re)deploy these components in their applications, in as many ways as possible. *Corollaries*:
  1. Bind the entire system together with a common, versatile scripting system to address a wide variety of mostly procedural "glue" requirements. (201)
  2. Make both the sets of component types and the communications system fully extensible so that functionality can be distributed across independently developed software libraries. (202)
  3. Provide both a rich set of basic information types for interchange and special, reusable non-component software services to broadly enhance component utility. (203)

The uniqueness of the current invention derives from the fact that it incorporates all of these features in one straightforward package.

The invention's software components can be arranged into nested or nestable network topologies formally referred to as *cybernets*, a contraction of the term *cybernetic*, found in branches of control theory, and *network*. The following definitions, borrowing from the